

Drupal 下如何把包含 Highcharts 的页面输出到 PDF

摘要

Highcharts 是一种通过 JavaScript 在浏览器客户端生成图的库，由于缺乏生成把 HTML 转换成 PDF 的 Javascript 库，生成 PDF 的工作只能放在服务器端做。但是服务器端如何比较完美地解释和渲染 HTML、Javascript、CSS 呢？答案在 PhantomJS。本文基于 Drupal 的权限和主题样式，讲述如何把这一系列的技术组合起来，实现包含 Highcharts 的页面输出到 PDF。

关键词：Drupal, Highcharts, PDF, PhantomJS, HTML, JavaScript, CSS

黄冠能

<hgneng at gmail.com>

2013 年 4 月



目录

1	什么是 Highcharts.....	3
2	Drupal 中如何使用 Highcharts.....	3
3	把 Highcharts 导出成 PDF 的若干种方法	4
3.1	利用 API 把单幅 Highcharts 导出成静态图再重新组合页面	4
3.2	html2canvas.....	4
3.3	wkhtmltopdf.....	4
3.4	PhantomJS	5
4	如何在 Drupal 实现权限控制.....	5
4.1	运行 phantomjs 时设置登录用户的 cookie.....	6
4.2	在页面 URL 中增加身份验证的参数	6
5	如何在 Drupal 为页面添加打印相关样式	7
6	打印的分页问题	7
7	PDF 字体问题	7
8	Highcharts 在 IE8 中渲染性能低下的问题	8
9	参考资料	8

1 什么是 Highcharts

Highcharts 是一种使用纯 HTML5/JavaScript 在浏览器端生成图表的 JavaScript 库，支持所有现代浏览器，如 Chrome、Firefox、IE、iPad/iPhone 上的浏览器 (Safari) 等。特别强调支持 IE6 (连 IE6 都支持，还有什么好担心的呢?)。全球前 100 强的企业中有超过 35 家使用了该技术。

2013 年 3 月 22 日，3.0 版本正式发布。

相比起在服务器后台生成图表的技术，在前端生成可以生成动态、有交互效果的图表。

对于非商业用途，Highcharts 是免费的，商业用途的费用可查询如下页面：
<http://shop.highsoft.com/highcharts.html>

以下是 Highcharts 的官方网站：<http://www.highcharts.com/>

2 Drupal 中如何使用 Highcharts

首先下载 Highcharts 库，放到 Drupal 的 sites/all/libraries 下，

然后在需要使用 Highcharts 的页面中添加如下一行，即可使用

```
drupal_add_js(libraries_get_path('highcharts') . '/js/highcharts.js');
```

余下就是参考 Highcharts 的 API 了，与 Drupal 无关。

Drupal 本身也有 Highcharts 模块，主要用于 Views 数据的图表展示：参考
<http://drupal.org/project/highcharts>

3 把 Highcharts 导出成 PDF 的若干种方法

3.1 利用 API 把单幅 Highcharts 导出成静态图再重新组合页面

Highcharts 本身提供导出成图片的方法（API 见 <http://docs.highcharts.com/#export-module>），但是它和整个页面的排版脱离，要把一幅幅 Highcharts 图导出后再和页面其它元素组合起来输出 PDF 工作量非常大。并且这也设计把 Highcharts 导出的静态图从外部（浏览器）传回 PHP。

Highcharts 提供了在服务器端导出静态图的方案（事实上它使用了我们之后会讨论的 PhantomJS 方案）：

<http://www.highcharts.com/component/content/article/2-news/52-serverside-generated-charts>

3.2 html2canvas

这是一个通过 JS 截屏的工具，但是它并非真正截屏，而是模拟。笔者并没有尝试使用，可以遇见，在不同浏览器，不同窗口大小会有不同效果，而且，我们希望 PDF 版会隐藏页面的一些元素（比如菜单、导航等）

软件主页：<http://html2canvas.hertzen.com/>

3.3 wkhtmltopdf

这是一个非常接近的解决方案，项目的目的很直接，就是在服务器端运行一个 Webkit 内核的命令，把 HTML 转换成 PDF。Drupal/PHP 里需要做的只是执行一条 system 语句。笔者一度以为这是最终的解决方案，但随着使用的深入，发现项目缺乏维护，遇到一个编码问题一直没有被修复，页面的渲染效果似乎也和 Chrome 有差异。它可能是基于一个比较旧的 Webkit 内核。鉴于以上原因笔者还是没有采用该方案。

软件主页：<https://code.google.com/p/wkhtmltopdf/>

3.4 PhantomJS

这是笔者最终采用的方案，它是一个基于 Webkit 的命令行可运行的浏览器，目前处于比较好的维护状态，它可以用于自动化测试。

软件主页：

<http://phantomjs.org/>

使用方法如下：

1. 把 PhantomJS 下载到服务器，并置于 \$PATH 中，笔者是做了一个符号链接到 /usr/bin/phantomjs
2. 创建如下 JS 脚本：generate_pdf.js

```
// This file is NOT a browser-run javascript but PhantomJS script

var system = require('system');
var html_path = system.args[1];
var pdf_path = system.args[2];

var page = require('webpage').create();
page.paperSize = {
  format: 'A4',
  orientation: 'landscape',
  border: '1cm'
};

page.open(html_path, function () {
  page.render(pdf_path);
  phantom.exit();
});
```

3. 运行如下命令即可把 HTML 页面输出到 PDF

```
$ phantomjs generate_pdf.js http://www.google.com google.pdf
```

使用 PhantomJS 生成的 PDF 比较大，我想它很可能每一页就是一张图片。

4 如何在 Drupal 实现权限控制

如果需要生成 PDF 的页面不能匿名访问，那么对于在命令行直接运行 phantomjs 是个问题，可以通过以下两种方式解决：

4.1 运行 phantomjs 时设置登录用户的 cookie

笔者并没有试验过这种方法，印象中相关 cookie 值在 Drupal 里的计算可能和用户来源有关，不确定直接把用户的 cookie 传给 phantomjs 是否能正常通过 Drupal 的身份验证。这种方法调试起来也比较麻烦。

4.2 在页面 URL 中增加身份验证的参数

这个方法的前提是该页面是自己写的，只有这样我们才能改变其权限验证以及页面的参数。具体方法是把用户的 uid 和 session ID 以参数形式传给页面，作为身份验证的参数，判断过程可参考以下代码：

```
// permission checking
if (!user_access('access content')) {
  $auth_uid = arg(4);
  $auth_sid = arg(5);
  if (isset($auth_uid) && isset($auth_sid)) {
    // check permission, used for authentication of print
    $account = user_load($auth_uid);
    if (user_access('access content', $account)) {
      $result = db_query('SELECT uid, sid
        FROM {sessions}
        WHERE uid = :uid AND sid = :sid',
        array(':uid' => $auth_uid, ':sid' => $auth_sid));
      if ($result->rowCount() == 0) {
        drupal_access_denied ();
        return;
      }
    } else {
      drupal_access_denied();
      return;
    }
  } else {
    drupal_access_denied();
    return;
  }
}
```

5 如何在 Drupal 为页面添加打印相关样式

首先，我们在页面中应该通过某种方式来区分当前正在生产的页面是正常浏览状态，还是处于生成 PDF 的状态。当处于生成 PDF 状态时，我们添加如下一行：
(假设我们使用 zen 主题，其它主题请按需修改其打印样式路径)

```
drupal_add_css(drupal_get_path('theme', 'zen') . '/css/print.css');
```

这样我们就能获得额外的打印渲染样式，print.css 中比较明显的效果是把内容区以外的东西都隐藏掉了。我们也可为页面的 PDF 版再通过 drupal_add_css 的方法添加额外的样式。

6 打印的分页问题

分页的样式是 page-break-after, page-break-before, page-break-inside, 如果我们在制定类的元素后分页可添加如下样式:

```
.bkaft {  
  page-break-after:always;  
}
```

然后在需要分页的元素中添加类:

```
<div class="bkaft">...</div>
```

参考: http://www.w3schools.com/cssref/pr_print_pageba.asp

值得注意的是，page-break*属性不能用于 position:absolute 和 float 元素，不能用在 table 和有边框的元素内。Chrome 分页成功的情况似乎要少于 IE。

7 PDF 字体问题

在部分 Linux 系统中，PhantomJS 可能无法渲染中文，这是和系统的字体有关。对于 Open SUSE Linux，我们可以使用以下命令解决:

```
$ sudo zypper install *font*
```

有些 Linux 系统的默认字体不好看，我们可以通过强制方式指定字体。下面的 "WenQuanYi Zen Hei" 是相对比较好的一中。page-xxx 需要根据页面具体情况修改。!important 是较少使用的 CSS 属性，它相当于把该样式设置为最高优先级，

非迫不得已不要使用，因为难以再写一些样式来覆盖它。

```
.page-xxx * {  
  font-family: "WenQuanYi Zen Hei" !important;  
}
```

如果想使用其它非系统自带字体（注意：这可能存在版权问题），可以把相关字体文件复制到/usr/share/fonts 目录下的一个自定义子目录，然后通过以下命令更新字体：

```
# cd /usr/share/fonts/windows  
# mkfontscale  
# mkfontdir  
# fc-cache
```

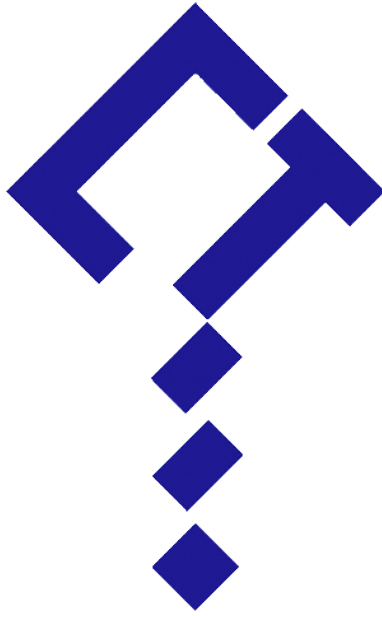
8 Highcharts 在 IE8 中渲染性能低下的问题

IE8 对于渲染 Highcharts 异常地慢，解决方案是让 IE8 在 IE7 兼容模式下运行（可能会快 10 倍），Drupal 中通过以下代码可实现：

```
$meta_ie_render_engine = array(  
  '#type' => 'html_tag',  
  '#tag' => 'meta',  
  '#attributes' => array(  
    'http-equiv' => 'X-UA-Compatible',  
    'content' => 'IE=7',  
  ),  
  '#weight' => -9999,  
);  
  
// Add header meta tag for IE to head  
drupal_add_html_head($meta_ie_render_engine, 'meta_ie_render_engine');
```

9 参考资料

1. Highcharts 主页：<http://www.highcharts.com/>
2. Drupal Highcharts 模块：<http://drupal.org/project/highcharts>
3. html2canvas 主页：<http://html2canvas.hertzen.com/>
4. wkhtmltopdf 主页：<https://code.google.com/p/wkhtmltopdf/>
5. PhantomJS 主页：<http://phantomjs.org/>



本资料由知未科技赞助制作
Sponsored By Curious-Tech.com